UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/686,882	10/15/2003	Kenneth Mark Williams	PA2594US	4103

22830 7590 02/13/2008

CARR & FERRELL LLP
2200 GENG ROAD
PALO ALTO, CA 94303

EXAMINER	
THOMAS, SHANE M	

ART UNIT	PAPER NUMBER
2186	

MAIL DATE	DELIVERY MODE
02/13/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/686,882	WILLIAMS ET AL.	
	Examiner	Art Unit	
	Shane M. Thomas	2186	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *06 November 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *2-6,17-20,24-34 and 37* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *2-5,17-20,24-34 and 37* is/are rejected.

7)☒ Claim(s) *6* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *6/29/07 & 9/6/07*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

This Office action is responsive to the amendment filed 11/6/2007. Claims 2-6, 17-20, 24-34 and 37 remain pending. Applicants' arguments have been carefully and respectfully considered, but they are not persuasive. Accordingly, this action has been made FINAL.

Response to Amendments/Arguments

Examiner has modified the prior art rejections in order to meet Applicant's amendments to the claims. Specifically, the previously cited Emma reference is being applied to teach the new limitation of "and extension adapter comprising a load/store unit" of claim 1, and the prior art reference of Roussel et al. is being applied to teach the new limitation of "execution of the unaligned data sequence" of claim 34. Additionally, Roussel et al. has been cited to teach claims 24-30. Finally, new matter rejections under 35 U.S.C. 112, first paragraph, have been presented as discussed below in regards to a portion of the newly claimed subject matter of claims 24-33.

Applicant argues on page 9 on the response, that the present invention as claimed distinguishes over the prior art of Peterson by the use of an instruction set extension fabric (ISEF) to enable instruction extensions to be stored and executed that are defined post-silicon. The Examiner has interpreted the term using a broadest reasonable approach and has applied the prior art reference of Emma to teach the amended claim limitation as discussed in the art rejections below. While the Applicant's specification and arguments state that the ISEF allows extended instructions to be added post silicon, such an amendment is not contained in the claims as currently presented and as such will not be addressed.

Claim Objections

Claims 18, 19, 26, and 31-33 are objected to because of the following informalities:

As per claims 18 and 19, the term --the one or more number of data sequences-- should be amended to the term --the one or more data sequences--.

As per claim 26, the Examiner recommends removing the underscore from the term "execution_time."

As per claim 31, the Examiner recommends amending claim 31, second to last line, to read: "aligned unaligned instruction output ~~to the memory~~ to create aligned data" to clarify the limitation.

Claims 32 and 33 are objected to as being dependent upon an objected base claim.

Appropriate correction is required.

Claim Rejections - 35 USC § 112

The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

Claims 2-6 and 24-34 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

As per claims 24 and 31, the Applicant does not teach the use of "executing a user defined instruction using the unaligned data." Applicant recites the steps of using load and store instructions in combination with a register file to retrieve and store data in a register file (¶38); however Applicant's originally-filed specification is silent with regard to the use of unaligned data by a user-defined instruction. Further, Applicant's originally-filed specification teaches a configuration memory (¶43) for use with user-defined instructions, but does not elaborate and discuss using an unaligned subset of data for one of the user-defined instructions. Further, the Applicant has not given the Examiner any guidance concerning where support for such a limitation can be found in the originally-filed specification. Applicant argues on page 9, paragraph 1, of the present response the term "unaligned data" has been used in the present claims to expedite prosecution and that the term has support in the specification as originally filed. While the Examiner agrees that support for the term "unaligned data" does indeed have support in the specification, the limitation of using the unaligned data by a user-defined instruction does not have proper support.

Likewise, concerning claim 31, no where in Applicant's originally filed specification can the Examiner determine where (1) a user-defined instruction is executed using the aligned data to create an unaligned instruction output; (2) receiving the unlighted output; (3) aligning unaligned instruction output to the memory to create aligned data; and (4) storing the aligned data in the memory.

As per claim 34, as stated above, Applicant's originally-filed specification does not provide support for the execution of an unaligned data sequence.

Claims 2-6, 25-30, 32, and 33 are rejected as being dependent upon a rejected base claim.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 24-30 are rejected under 35 U.S.C. 102(e) as being anticipated by Roussel et al. (U.S. Patent Application Publication No. 2003/0120889).

As per claim 24, Roussel teaches:

A method of processing data, the method comprising:

receiving a sequence of aligned data at an extension adapter (the examiner is considering the --extension adapter-- to be the FPU unit 56 of the processor 10, where the load converter 57 of the FPU 56 receives the aligned data before shifting and merging the data into an unaligned operand - ¶28). Roussel teaches in ¶26 that a sequence of aligned data (e.g. the operand data from two uop loads) may be retrieved from a cache 12 of the processor 10.

the receipt of the sequence controlled by a processor using a load/store instruction - ¶26 (the load instruction "LDDDQU," when executed by the processor, initiates the two uop loads of aligned data to the cache 12);

receiving a user defined instruction at the extension adapter from the processor (the processor 10 can execute floating point instructions {e.g. user-defined since a user/programmer can select instructions a processor is to execute} - ¶25; therefore, in order to execute a floating point instruction, the FPU 56 {e.g. extension adapter} is utilized- ¶25);

loading unaligned data which is a subset of the sequence of aligned data into a programmable instruction set extension fabric - ¶¶28-29 (unaligned data is loaded into an XMM register {e.g. the XMM entire register set of the execution environment shown in figure 2 is considered to be the --extension fabric--} after it is created via shifting and merging performed by the load converter 57 of the FPU 56)**; and**

executing the user defined instruction using the unaligned data (once stored in the XMM registers 84, the FPU unit may execute floating point or SIMD user-defined instruction using the unaligned data - ¶19 and ¶23)**.**

As per claim 25, Roussel teaches:

stalling execution of the user defined instruction to assure availability of the unaligned data - ¶¶26-29 (it can be seen that any instruction that requires an unaligned operand to execute will be stalled or prevented from execution until the uloads {¶26} and the shifting and merging steps {¶27} are completed and the unaligned operand stored in the designated XMM register {¶29}).

As per claim 26, Roussel teaches:

controlling the timing of the loading of the unaligned data a function of instruction execution time - ¶¶26-29 (as discussed above with respect to claim 25, ¶¶26-29 teach that the timing of the loading of the unaligned data occurs after the loading, shifting, and merging steps occurring to the aligned data in order to produce the unaligned data; thus, the **time of execution** of the under-defined instruction must occur after a time when the unaligned data {e.g. the operand which is used by the instruction} is available in the XMM register).

As per claim 27, Roussel teaches:

operating on both aligned and unaligned data using the programmable instruction set extension fabric (unaligned operation using the ISEF has be discussed *supra* with regard to claim 24 while Roussel teaches operation of aligned data using the ISEF {e.g. as defined, the set of all registers of the execution environment of figure 2} in ¶19 and specifically the execution of integer {e.g. aligned} operands in ¶25, second column)

the alignment being relative to a word size of the processor (the alignment is relative to a word size as discussed in ¶27).

As per claim 28, Roussel teaches:

wherein loading the unaligned data of the sequence into the programmable instruction set extension fabric is accomplished using an instruction configured to read data of a first size and to increment a memory address pointer by an amount different from the first size (the "LDDQU" instruction loads, shifts, and merges aligned data to create an unaligned data which is then stored in the ISEF XMM register - ¶29). It can be seen that the because the uload caused by the LDDQU instruction (¶28) causes a shift of less than the whole aligned operand size of each of the aligned operands used to create the unaligned operand, that the amount to be shifted is being considered to be the amount to increment a memory address pointer by to begin reading the valid portion of the unaligned operand contained in the first of the aligned operands.

As per claim 29, Roussel teaches:

initializing a data pointer in order to achieve an offset between a data alignment of the processor and the unaligned data (the amount bits to shift during the shifting of the aligned operands {¶28} is being considered to be a --data pointer-- as the shifting is responsible for

creating an offset between the data alignment of the aligned operands of the processor and the unaligned operands - ¶¶26-27).

As per claim 30, Roussel teaches:

wherein loading unaligned data which is a subset of the sequence of aligned data (as discussed *supra* with regard to claim 24 above) **is performed using a PUT instruction received from the processor** (the Examiner is considering the "LDDQU" instruction sent from the processor to be a --PUT instruction-- as the LDDQU instruction creates and stores the unaligned data operand in the XMM register - ¶¶26-29, which "puts" the unaligned data into the ISEF XMM register).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 17-20 and 37 are rejected under 35 U.S.C. 103(a) as being unpatentable by Petersen (U.S. Patent No. 5,517,627) in view of Ramagopal et al. (U.S. Patent Application Publication No. 2003/0196058) in further view Emma (U.S. Patent No. 5,619,665). Further the prior art references of Voith et al. (U.S. Patent No. 5,636,224) and Tanenbaum ("Modern Operating Systems") are being used to support an inherent feature of the Petersen and Ramagopal et al. references, respectively.

In order to provide a coherent rejection, the Examiner will follow claim dependence following each independent claim when discussing the claim rejections. For example, the Examiner will first discuss the claim rejections for independent claim 17 and all dependent claims before discussing independent claim 34 and its respective dependent claims.

As per claim 17, Petersen teaches a **system** (figures 7-8) comprising a **processor** 10 **configured to execute one or more data sequences** (e.g. instructions as it is inherent in the art that a processor execute instructions); **an extension adapter** 15 **including a load/store buffer** 24, **the extension adapter configured** (via the **load/store buffer** 24) **to store data** (refer to figures 4A-4C and 5A-5B where it is shown that data bytes are stored in provided registers) and a **processor** 10 **configured to execute GET** (read) **instructions** (write and read aligners can be implemented with a ROM, which is well known in the art to contain instructions for controlling hardware systems, - [13/7-11]) for **processing data sequences.**

Alternatively, the Examiner could consider the memory controller 26 as being a --processor-- as the memory controller can be seen in regard to figure 8 of Petersen to interact with the buffer memory 27 and the load/store buffer 24. Since the instructions to load/store the buffer 24 may be implemented via a ROM [13/8-11], it is necessarily inherent that some element perform the operation of executing the instructions (e.g. data sequences) stored on the ROM in order to utilize the load/store buffer 24. That element is being considered by the Examiner to be the --processor--.

Petersen teaches **initializing a load/store buffer** (data aligner 24, figure 8) **by loading a first aligned word** (4 bytes - [7/55-63]) **into the load/store buffer** (figure 5A - [8/41-47]) **from memory** 27 - [12/18-20]. The Examiner is considering loading a data word from the buffer 27 to

the data aligner 24 as --initializing-- since the process places data into the data aligner. It can be seen with respect to figures 5A and 5B that it takes two loads to the data aligner 24 from the buffer memory 27 to --initialize-- the data aligner by placing data therein.

Further, Petersen teaches **further initializing the load store buffer** 24 **by loading a second aligned word** (figure 5B) **into the load/store buffer** [8/48-50] **from the memory**.

While Peterson **teaches reading one or more data sequences** (wherein a single data sequence is being defined by the Examiner as the number of bits transferred per lane L2(i), in the present embodiment of Petersen - one byte (8 bits) - [7/55-63]) **from the load/store buffer** 24, the reference does not specifically teach reading those data sequences **into a register file for instruction execution**. Peterson teaches that a DMA controller 13 is used to issue read and write requests on the host bus 14 to the peripheral device 15, thereby alleviating host processor from performing such duties [9/49 - 10/2], in addition to sending data from the system memory 12 to the peripheral device 15 [9/62-66]. The data retrieved or written to the buffer 27 is sent to the host bus 14 - [12/18-20].

Register files are well known in the processing and memory arts to be the collection of local processor registers **configured for instruction execution** (see noted prior art of record but not relied upon in the Conclusion section of this action).

Ramagopal teaches in ¶¶30-31 that data is received and sent between the memory 12 and register file 28 for instruction execution and storage after execution instruction, respectively (figure 1). Ramagopal further teaches that DMA accesses may be used to access memory 12 to store data and the like between the memory 12 and an external source (¶35). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to

have combined the load/store buffering unit of Petersen with the teaching of associating a DMA access (DMA to/from external source to/from a memory that interfaces the register file) with a register file of Ramagopal in order to have (1) enabled the processor 10 of Petersen to have buffered the data to be executed in a memory (register file) very close in relation to the processor (to avoid having to continually fetch the data for each execution from the memory 12 during every instruction execution, thereby increasing instruction throughput) and (2) utilized the DMA processing to quickly send and receive data between the memory 12 and an external source (for example the peripheral device 15 of Petersen) for forwarding on to the register file - ¶35 of Ramagopal.

As a result of the combination of Petersen and Ramagopal, it can be seen that the **data sequences** from the **load/store buffer** 24 of modified Petersen would have been **read into the register file** (through the DMA controller's interaction between the memory 12 via the host bus 14 and the external device 15, comprising the load/store buffer 24), where the data sequences would have been **configured for instruction execution** - ¶35 of Ramagopal.

Further, as is known in the art of DMA accessing, the **reading of the one or more sequences from the load/store buffer into a register file** may be performed using an instruction receiving from the processor. The Examiner is considering this --instruction-- to be the number of bytes to transfer and the memory addresses involved that is commonly sent from the processor to the DMA device in order to initiate the DMA. Tanenbaum teaches such well-known DMA operation in lines 1-5 of page 31.

Peterson further teaches **at least one of the one or more data sequences being unaligned relative to the memory** [7/52-54].

Finally, Peterson teaches **loading additional aligned words to the load/store buffer from the memory** 27 **to replace the one or more data sequences that are read from the load/store buffer into the register file** ([13/51/54], where the reading process can be repeated as necessary to read data from the buffer memory 27 - [figure 13, step 153 - step 142]). The data sequences can be read into the register file as discussed above with reference to the Ramagopal teachings.

Modified Peterson does not specifically teach but Emma does teach a **programmable instruction set extension fabric** (figure 5, elements 506 and 504) **for enabling instruction extensions to be stored and executed** (instruction extensions can be stored in the EI-Cache 50 and sent to the processor 508 for execution [11/42-62]). Emma teaches in the abstract that extensions of an instruction set allows for circumventing software compatibility issues when allowing legacy software to benefit from new architectural extensions without recompilation and reassembly. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to have combined the data aligner system of figure 7 of Petersen with the teaching of an extension instruction set and the programmable instruction set extension fabric of Emma in order to have gained expandability of the instruction set of the processor 10 of Petersen, thereby gaining flexibility and compatibility for future instruction extensions without having to recompile or reassemble the system of Petersen.

As per claim 18, Petersen teaches that **the number of data sequences read is an immediate specified number** that is specified by the host [8/62-67].

As per claim 19, Petersen teaches that **the number of data sequences read is a specified number stored as an index in a register memory** [9/4-6]. The Examiner is considering the value of the "CURRENT READ" to be an index.

As per claim 20, Petersen teaches **in which a first one of the one or most data sequences read is located at a first memory location** (i.e. any of the bytes of the 4 byte word entry of FIFO buffer 27 that is first read into the data aligner upon a read request from the host - [8/41-46] that initializes the data aligner as discussed supra) **and the one or more data sequences comprises a specified number of data sequences stored in a register memory** 27 (here the Examiner is considering the buffer memory 27 as being the register memory and the --specified number-- of data sequences stored at the first index of the [FIFO] buffer 27 is specified to be four - as each [FIFO] buffer 27 entry contains four data sequences (four aligned bytes) - [1/26-35] and [8/30-50] - where the --first index-- is the index of the first entry of the FIFO buffer 27 to be read), **wherein the subsequent data sequence following the first of the data sequences is located at a second memory location pointed to by a second index** (the next memory location that is accessed from the FIFO buffer 27 - [8/48-50] as taught in the following example).

Petersen teaches that the first data sequence read can be any of the data sequences of the first FIFO buffer, with data unit 01 being used as an example that is read first [8/41-47]. If for example, the 04 data sequence was read first (i.e. the --first of the one or more data sequences--), thereby leaving data sequences 01,02, and 03, in the queues of the data aligner, data sequences

01, 02, 03, 05, 06, 07, and 08, would have been available to the host [8/51-52]. Thus it can be seen that the **subsequent data sequence** (data sequence 05) **following the first of the data sequences** (04) **is located at a second memory location** (buffer 27, second entry) that is **pointed to by a second index** (i.e. whatever the index/address of the FIFO buffer 27 contains the next data word that is read after the first data word - [8/48-50]).

As per claim 37, Peterson teaches **wherein the length of at least one of the one or more data sequences read from the load/store unit into the register file is less than a length of the first aligned word** (figures 5A-5B and [8/41-47]). The cited passages teach an example of reading a data length of one byte (e.g. data unit 01) from the load/store buffer 24, which is less than the length of the first aligned work (e.g. four bytes) - [7/60-63].

Claims 2,3,5,6 and 34 are rejected under 35 U.S.C. 103(a) as being unpatentable by Petersen (U.S. Patent No. 5,517,627) in view of Ramagopal et al. (U.S. Patent Application Publication No. 2003/0196058) in further view of Roussel et al. (U.S. Patent Application Publication No. 2003/0120889). Further the prior art references of Voith et al. (U.S. Patent No. 5,636,224) and Tanenbaum ("Modern Operating Systems") are being used to support an inherent feature of the Petersen and Ramagopal et al. references, respectively.

As per claim 34, the art rejection as well as the motivation to combine the Peterson and Ramagopal references follows the rejection and motivation set forth *supra* with regard to claim 17.

Further, Peterson teaches that the load/store buffer is **in an extension adapter** 15, figures 7-8, as well as **the alignment of the first and second word are relative to a memory** 27

accessible to the processor 10 - which is 4 data units [7/60-63], [8/42-47], [12/18-20], and [14/5-8]. Peterson additionally teaches **the unaligned data sequence including at least a part of the second aligned word** as taught in [8/52-55] with regard to the example posed in figures 5A-5B. Here, byte **05**, which is part of the second aligned word as shown in figure 5B, may be read if the processor requests a full width read (4 data units).

Modified Peterson does not specifically teach **executing the unaligned data sequence**. Roussel teaches execution of operands (e.g. unaligned data sequences) in ¶25 and teaches that the operands may be unaligned in ¶¶26-27. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to have combined the unaligned accessing system of modified Peterson with the teaching of executing the unaligned data sequences, once retrieved from a memory, in order to have achieved the predictable result of instruction execution using unaligned data sequences. Both the Peterson and the Roussel reference are in the same field of endeavor - unaligned accessing.

As per claim 2, Petersen teaches that **the data sequence length** [read] **is a byte** [8/42-43] and [7/60-61].

As per claim 3, Petersen teaches that **the data sequence length** [read] **is a bit** [7/64-67].

As per claim 5, Peterson teaches **wherein each of the one of the one or more unaligned data sequences has a length less than a length of the second aligned word** (figures 5A-5B and [8/41-47]). The cited passages teach an example of reading a data length of one byte (e.g. data unit 01) from the load/store buffer 24, which is less than the length of the first aligned work (e.g. four bytes) - [7/60-63]. It can be seen with reference to figures 5A-5B and [8/30-50] that

the length of the second aligned word is the same length (e.g. 4 data units) as the first aligned word.

Claim 4 is rejected under 35 U.S.C. 103(a) as being unpatentable by Petersen (U.S. Patent No. 5,517,627) in view of Ramagopal et al. (U.S. Patent Application Publication No. 2003/0196058) in further view of Roussel et al. (U.S. Patent Application Publication No. 2003/0120889), as applied to claim 34 above, in further view of Emma (U.S. Patent No. 5,619,665). Additionally, the prior art references of Voith et al. (U.S. Patent No. 5,636,224) and Tanenbaum ("Modern Operating Systems") are being used to support an inherent feature of the Petersen and Ramagopal et al. references, respectively.

As per claim 4, Peterson teaches a general-purpose processor (host processor 10 - figure 7) and Emma teaches **an extensible instruction set** (abstract). Emma teaches in the abstract that extensions of an instruction set allows for circumventing software compatibility issues when allowing legacy software to benefit from new architectural extensions without recompilation and reassembly. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to have combined the data aligner system of figure 7 of Petersen with the teaching of an extension instruction set and the programmable instruction set extension fabric of Emma in order to have gained expandability of the instruction set of the processor 10 of Petersen, thereby gaining flexibility and compatibility for future instruction extensions without having to recompile or reassemble the system of Petersen.

Allowable Subject Matter

Claim 6 would be allowable if rewritten to overcome the rejection(s) under 35 U.S.C. 112, 2nd paragraph, set forth in this Office action and to include all of the limitations of the base claim and any intervening claims.

The following is a statement of reasons for the indication of allowable subject matter:

As per claim 6, the prior art of record does not teach nor reasonably suggest, either alone or in combination, changing a memory address pointer by an amount that is less than the length of the second aligned word in order to point to the next unaligned data sequence that is to be read.

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Shane M. Thomas whose telephone number is (571) 272-4188. The examiner can normally be reached M-F 8:30 - 5:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Matt M. Kim can be reached at (571) 272-4182. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Shane M. Thomas

MATTHEW KIM
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100